



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+many +icon execut\$ copyright



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **many icon execut\$ copyright**

Found 5,245 of 167,655

Sort results by

relevance

[Save results to a Binder](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)

Display results

expanded form

[Search Tips](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1 [Fast detection of communication patterns in distributed executions](#)**

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available: [pdf\(4.21 MB\)](#) . Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

**2 [A recursive interpreter for the Icon programming language](#)**

J. O'Bagy, R. E. Griswold

July 1987 **ACM SIGPLAN Notices , Papers of the Symposium on Interpreters and interpretive techniques SIGPLAN '87**, Volume 22 Issue 7

Publisher: ACM Press

Full text available: [pdf\(923.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The implementation of the Icon programming language is more interesting and difficult than the implementation of many other programming languages because an expression in Icon can generate a sequence of results. The implementation therefore must support control backtracking in expression evaluation. There also are several novel control structures related to generators. Because expression evaluation is limited lexically, a full coroutine mechanism is not needed and expression evaluation can be ha ...

**3 [Human-computer interface development: concepts and systems for its management](#)**

H. Rex Hartson, Deborah Hix

March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Publisher: ACM Press

Full text available: [pdf\(7.97 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

*Human-computer interface management*, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

#### 4 Interactive Editing Systems: Part II



Norman Meyrowitz, Andries van Dam

September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

**Publisher:** ACM Press

Full text available: [pdf\(9.17 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 5 Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers



Caitlin Kelleher, Randy Pausch

June 2005 **ACM Computing Surveys (CSUR)**, Volume 37 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(14.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Since the early 1960's, researchers have built a number of programming languages and environments with the intention of making programming accessible to a larger number of people. This article presents a taxonomy of languages and environments designed to make programming more accessible to novice programmers of all ages. The systems are organized by their primary goal, either to teach programming or to use programming to empower their users, and then, by each system's authors' approach, to make ...

**Keywords:** Human-computer interaction, computer Science education, learning, literacy, problem solving

#### 6 Digital libraries, value, and productivity



Gio Wiederhold

April 1995 **Communications of the ACM**, Volume 38 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(292.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A digital library is popularly viewed as an electronic version of a public library. But replacing paper by electronic storage leads to three major differences: storage in digital form, direct communication to obtain material, and copying from a master version. These differences in turn lead to a plethora of further differences, so that eventually the digital library no longer mimics the traditional library. Furthermore, a library is only one element in the process of creating, storing, culling, and ...

#### 7 Generators in Icon



Ralph E. Griswold, David R. Hanson, John T. Korb

April 1981 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 3 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(1.03 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 8 Late breaking results: posters: Predicting task execution time on handheld devices using the keystroke-level model



Lu Luo, Bonnie E. John

April 2005 **CHI '05 extended abstracts on Human factors in computing systems**

**Publisher:** ACM Press

Full text available: [pdf\(310.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Keystroke-Level Model (KLM) has been shown to predict skilled use of desktop systems, but has not been validated on a handheld device that uses a stylus instead of a keyboard. This paper investigates the accuracy of KLM predictions for user interface tasks running on a Palm OS based handheld device. The models were produced using a recently

developed tool for KLM construction, CogTool, and were compared to data obtained from a user study of 10 participants. Our results have shown that the KL ...

**Keywords:** cognitive modeling, handheld device, keystroke-level model, task execution time

9 A preferable look—APL in window-based environments



Ursula Recker, Michael Rys

May 1990 **ACM SIGAPL APL Quote Quad , Conference proceedings on APL 90: for the future APL '90**, Volume 20 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(968.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Currently, most APL systems are line and workspace oriented. It will be shown how a modern APL could fit into a window-based, event driven environment. To achieve this goal, a future user interface in an experimental environment is sketched by presenting its language environment, the editor and the debugging facility.

10 Computing curricula 2001



September 2001 **Journal on Educational Resources in Computing (JERIC)**

**Publisher:** ACM Press

Full text available: [pdf\(613.63 KB\)](#) [html\(2.78 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Expression evaluation in the icon programming language



Ralph E. Griswold

August 1984 **Proceedings of the 1984 ACM Symposium on LISP and functional programming**

**Publisher:** ACM Press

Full text available: [pdf\(507.75 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 SCIL-VP: a multi-purpose visual programming environment



Dennis Koelma, Richard van Balen, Arnold Smeulders

March 1992 **Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's**

**Publisher:** ACM Press

Full text available: [pdf\(1.21 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

13 Internet-based workflows: a paradigm for dynamically reconfigurable desktop environments



Hemang Lavana, Amit Khetawat, Franc Brglez

November 1997 **Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge**

**Publisher:** ACM Press

Full text available: [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** Internet, Petri net, collaborative, desktop, reconfigurable, recordable, workflows

14 Ubiquity: Making sense of sensing systems: five questions for designers and researchers



Victoria Bellotti, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter, Austin Henderson, Cristina Lopes

April 2002 **Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves**

**Publisher:** ACM Press

Full text available: [pdf\(435.34 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper borrows ideas from social science to inform the design of novel "sensing" user-interfaces for computing technology. Specifically, we present five design challenges inspired by analysis of human-human communication that are mundanely addressed by traditional graphical user interface designs (GUIs). Although classic GUI conventions allow us to finesse these questions, recent research into innovative interaction techniques such as 'Ubiquitous Computing' and 'Tangible Interfaces' has begun ...

**Keywords:** design framework, human-machine communication, sensing input, social science, ubiquitous computing

15 A high-level programming and command language



Christopher W. Fraser, David R. Hanson

June 1983 **Proceedings of the 1983 ACM SIGPLAN symposium on Programming language issues in software systems**

**Publisher:** ACM Press

Full text available: [pdf\(658.67 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Unifying programming and command languages is a promising idea that has yet to be thoroughly exploited. Most attempts at such unification have used Lisp or traditional languages, such as Pascal. This paper describes the command and programming language EZ, which attempts to unify command and programming languages by using high-level string-processing concepts, such as those in SNOBOL4 and Icon. EZ has particularly simple data abstractions that attempt to bring ...

16 Software watermarking: models and dynamic embeddings



Christian Collberg, Clark Thomborson

January 1999 **Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

**Publisher:** ACM Press

Full text available: [pdf\(2.19 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 GLEAN: a computer-based tool for rapid GOMS model usability evaluation of user interface designs



David E. Kieras, Scott D. Wood, Kasem Abotol, Anthony Hornof

December 1995 **Proceedings of the 8th annual ACM symposium on User interface and software technology**

**Publisher:** ACM Press

Full text available: [pdf\(1.15 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** GOMS models, software and technology, usability, usability evaluation, user models, user-interface

18 Late breaking result papers: Augmenting icons for deaf computer users



Helen Petrie, Wendy Fisher, Kurt Weimann, Gerhard Weber



April 2004 **CHI '04 extended abstracts on Human factors in computing systems**

**Publisher:** ACM Press

Full text available: [pdf\(231.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Tooltips (TTs) can be used to make icons more understandable to users. However, text-based tooltips will not assist users with print disabilities. Four types of TTs to assist deaf and hearing impaired users were implemented: Sign Language, Picture (an enlarged icon and text explanation of the function), Human Mouth and Digital Lips (the last two to assist in lip reading). An evaluation of 16 TTs of each type with 15 deaf users found that the Sign Language and Picture TTs were very positively rat ...

**Keywords:** adaptation, deafness, time design, tooltips

## 19 A relational approach to monitoring complex systems



Richard Snodgrass

May 1988 **ACM Transactions on Computer Systems (TOCS)**, Volume 6 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(3.42 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Monitoring is an essential part of many program development tools, and plays a central role in debugging, optimization, status reporting, and reconfiguration. Traditional monitoring techniques are inadequate when monitoring complex systems such as multiprocessors or distributed systems. A new approach is described in which a historical database forms the conceptual basis for the information processed by the monitor. This approach permits advances in specifying the low-level data collection, ...

## 20 Denotational semantics of a goal-directed language



David A. Gudeman

January 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 1

**Publisher:** ACM Press

Full text available: [pdf\(1.33 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Goal-directed evaluation is a very expressive programming language paradigm that is supported in relatively few languages. It is characterized by evaluation of expressions in an attempt to meet some goal, with resumption of previous expressions on failure. This paradigm is found in SNOBL4 in its pattern-matching facilities, and in Icon as a general part of the language. This paper presents a denotational semantics of Icon and shows how Icon is in fact a combination of two distinct paradigms ...

**Keywords:** generators, goal-directed evaluation, programming language paradigms

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright ?2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+many +icon execut\$ copyright



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used many icon execut\$ copyright

Found 5,245 of 167,655

Sort results by Display results  [Save results to a Binder](#) [Search Tips](#)☐ [Open results in a new window](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)Results 21 - 40 of 200 Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**21** [State trees as structured finite state machines for user interfaces](#)

James Rumbaugh

January 1988 **Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software**

Publisher: ACM Press

Full text available: [pdf\(1.77 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

State trees are a technique for specifying the control of an interactive system by organizing states into trees to define shared structure and behavior. The tree structure permits inheritance of state information, event traps, and entry and exit actions from states to substates, thereby sharing information and reducing the amount of code to be written. An interface can be restructured by moving entire subtrees as modules. State trees separate the recognition of commands from their implement ...

**22** [Lightweight lexical source model extraction](#)

Gail C. Murphy, David Notkin

July 1996 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 5 Issue 3

Publisher: ACM Press

Full text available: [pdf\(364.49 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#), [review](#)

Software engineers maintaining an existing software system often depend on the mechanized extraction of information from system artifacts. Some useful kinds of information—source models—are well known: call graphs, file dependences, etc. Predicting every kind of source model that a software engineer may need is impossible. We have developed a lightweight approach for generating flexible and tolerant source model extractors from lexical specifications. The approach is lightweight ...

**Keywords:** lexical analysis, lexing, reverse engineering, scanner generation, scanning, software maintenance, source code analysis, source model, static analysis

**23** [System support for pervasive applications](#)

Robert Grimm, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, David Wetherall

November 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 4

Publisher: ACM Press

Full text available: [pdf\(1.82 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Pervasive computing provides an attractive vision for the future of computing. Computational power will be available everywhere. Mobile and stationary devices will dynamically connect and coordinate to seamlessly help people in accomplishing their

tasks. For this vision to become a reality, developers must build applications that constantly adapt to a highly dynamic computing environment. To make the developers' task feasible, we present a system architecture for pervasive computing, called & ...

**Keywords:** Asynchronous events, checkpointing, discovery, logic/operation pattern, migration, one.world, pervasive computing, structured I/O, tuples, ubiquitous computing

## 24 Extending a database system with procedures



Michael Stonebraker, Jeff Anton, Eric Hanson

September 1987 **ACM Transactions on Database Systems (TODS)**, Volume 12 Issue 3

**Publisher:** ACM Press

Full text available: pdf(2.15 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper suggests that more powerful database systems (DBMS) can be built by supporting database procedures as full-fledged database objects. In particular, allowing fields of a database to be a collection of queries in the query language of the system is shown to allow the natural expression of complex data relationships. Moreover, many of the features present in object-oriented systems and semantic data models can be supported by this facility. In order to implement this cons ...

## 25 Planning and user interface affordances



Robert St. Amant

December 1998 **Proceedings of the 4th international conference on Intelligent user interfaces**

**Publisher:** ACM Press

Full text available: pdf(1.07 MB)

Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** affordances, planning

## 26 Dynamic metrics for java



Bruno Dufour, Karel Driesen, Laurie Hendren, Clark Verbrugge

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

**Publisher:** ACM Press

Full text available: pdf(222.67 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In order to perform meaningful experiments in optimizing compilation and run-time system design, researchers usually rely on a suite of benchmark programs of interest to the optimization technique under consideration. Programs are described as *numeric*, *memory-intensive*, *concurrent*, or *object-oriented*, based on a qualitative appraisal, in some cases with little justification. We believe it is beneficial to quantify the behaviour of programs with a concise and precisely ...

**Keywords:** Java, dynamic metrics, execution traces, optimization, profiling, program analysis, software metrics

## 27 Supporting concurrency, communication, and synchronization in human-computer interaction—the Sassafras UIMS



Ralph D. Hill

July 1986 **ACM Transactions on Graphics (TOG)**, Volume 5 Issue 3

**Publisher:** ACM Press

Full text available: pdf(2.44 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

[terms, review](#)

Sassafras is a prototype User Interface Management System (UIMS) specifically designed to support a wide range of user interface styles. In particular, it supports the implementation of user interfaces where the user is free to manipulate multiple input devices and perform several (possibly related) tasks concurrently. These interfaces can be compactly represented and efficiently implemented without violating any of the rules of well-structured programming. Sassafras also supports elaborate ...

## 28 The X window system



Robert W. Scheifler, Jim Gettys

April 1986 **ACM Transactions on Graphics (TOG)**, Volume 5 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(2.76 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

An overview of the X Window System is presented, focusing on the system substrate and the low-level facilities provided to build applications and to manage the desktop. The system provides high-performance, high-level, device-independent graphics. A hierarchy of resizable, overlapping windows allows a wide variety of application and user interfaces to be built easily. Network-transparent access to the display provides an important degree of functional separation, without significantly affecting ...

## 29 ITS: a tool for rapidly developing interactive applications



Charles Wiecha, William Bennett, Stephen Boies, John Gould, Sharon Greene

July 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 3

**Publisher:** ACM Press

Full text available: [pdf\(2.61 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The ITS architecture separates applications into four layers. The action layer implements back-end application functions. The dialog layer defines the content of the user interface, independent of its style. Content specifies the objects included in each frame of the interface, the flow of control among frames, and what actions are associated with each object. The style rule layer defines the presentation and behavior of a family of interaction techniques. Finally, the style program layer i ...

## 30 Pen computing: a technology overview and a vision



André Meyer

July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

**Publisher:** ACM Press

Full text available: [pdf\(5.14 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions. Starting with a short historic ...

## 31 An Alternative to the Use of Patterns in String Processing



Ralph E. Griswold, David R. Hanson

April 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 2 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(1.25 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

SNOBOL4 is best known for its string processing facilities, which are based on patterns as data objects. Despite the demonstrated success of patterns, there are many shortcomings associated with their use. The concept of patterns in SNOBOL4 is examined and problem



areas are discussed. An alternative method for high-level string processing is described. This method, implemented in the programming language Icon, employs generators, which are capable of producing alternative v ...

### 32 The Evaluation of Expressions in Icon



Ralph E. Griswold

October 1982 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 4 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

### 33 Engineering a simple, efficient code-generator generator



Christopher W. Fraser, David R. Hanson, Todd A. Proebsting

September 1992 **ACM Letters on Programming Languages and Systems (LOPLAS)**,  
Volume 1 Issue 3

**Publisher:** ACM Press

Full text available: [pdf\(853.86 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Many code-generator generators use tree pattern matching and dynamic programming. This paper describes a simple program that generates matchers that are fast, compact, and easy to understand. It is simpler than common alternatives: 200–700 lines of Icon or 950 lines of C versus 3000 lines of C for Twig and 5000 for burg. Its matchers run up to 25 times faster than Twig's. They are necessarily slower than burg's BURS (bottom-up rewrite system) matchers, but they are more flexible and s ...

**Keywords:** Icon programming language, code generation, code-generator generator, dynamic programming, tree pattern matching

### 34 Personal computer networks and graphical animation: Rationale and practice for education



Marc Brown, Norman Meyrowitz

February 1983 **ACM SIGCSE Bulletin , Proceedings of the fourteenth SIGCSE technical symposium on Computer science education SIGCSE '83**, Volume 15 Issue 1

**Publisher:** ACM Press

Full text available: [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper examines how progress in computer hardware and software may be applied to solve several serious problems in teaching computer science courses. It is concerned primarily with two such problems: 1) the lack of immediate reinforcement of computing concepts because of long delays between learning and practice, and 2) the difficulty instructors have motivating and explaining complex topics with currently available instruction tools and techniques. The paper first reviews the involve me ...

### 35 Copyright in shareware software distributed on the Internet—the Trumpet Winsock case



Cristina Cifuentes, Anne Fitzgerald

May 1997 **Proceedings of the 19th international conference on Software engineering**

**Publisher:** ACM Press

Full text available: [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** Internet service provider, copyright, distribution, intellectual property, shareware

36 PRIME—toward process-integrated modeling environments: 1

Klaus Pohl, Klaus Weidenhaupt, Ralf Dömges, Peter Haumer, Matthias Jarke, Ralf Klamma  
October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**,  
Volume 8 Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(1.15 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Research in process-centered environments (PCEs) has focused on project management support and has neglected method guidance for the engineers performing the (software) engineering process. It has been dominated by the search for suitable process-modeling languages and enactment mechanisms. The consequences of process orientation on the computer-based engineering environments, i.e., the interactive tools used during process performance, have been studied much less. In this article, we prese ...

**Keywords:** PRIME, method guidance, process modeling, process-centered environments, process-integrated environments, process-sensitive tools, tool integration, tool modeling

37 Using GOMS for user interface design and evaluation: which technique?

Bonnie E. John, David E. Kieras  
December 1996 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 3  
Issue 4

**Publisher:** ACM Press

Full text available: [pdf\(272.60 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Since the seminal book, The Psychology of Human-Computer Interaction, the GOMS model has been one of the few widely known theoretical concepts in human-computer interaction. This concept has spawned much research to verify and extend the original work and has been used in real-world design and evaluation situations. This article synthesizes the previous work on GOMS to provide an integrated view of GOMS models and how they can be used in design. We briefly describe the major ...

**Keywords:** GOMS, cognitive modeling, usability engineering

38 A technique for monitoring run-time dynamics of an operating system and a microprocessor executing user applications

Pramod V. Argade, David K. Charles, Craig Taylor  
November 1994 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the sixth international conference on Architectural support for programming languages and operating systems ASPLOS-VI**, Volume 29 , 28 Issue 11 , 5

**Publisher:** ACM Press

Full text available: [pdf\(978.78 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we present a non-invasive and efficient technique for simulating applications complete with their operating system interaction. The technique involves booting and initiating an application on a hardware development system, capturing the entire state of the application and the microprocessor at a well defined point in execution and then simulating the application on microprocessor simulators. Extensive statistics generated from the simulators on run-time dynamics of the applic ...

39 Evaluating mobile content: Cultural difference and mobile phone interface design: icon recognition according to level of abstraction

Ji Hye Kim, Kun Pyo Lee  
September 2005 **Proceedings of the 7th international conference on Human computer interaction with mobile devices & services MobileHCI '05**

**Publisher:** ACM Press

Full text available: [pdf\(944.96 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Mobile phone market has widened to a global scale and consequently mobile phones are distributed throughout the world. This tells that the user interface in mobile phones inevitably confronts cultural difference as much as other products and consequently the user interface suited to each cultural trait is required. To clarify the relation between cultural traits and mobile phone interface, UI elements which would be influenced by cultural traits in interaction between user and mobile phone were ...

**Keywords:** cultural difference, icon recognition, mobile phone interface

40 Communicating with icons as computer commands



Philip Rubens, Robert Krull

October 1988 **Proceedings of the 6th annual international conference on Systems documentation**

**Publisher:** ACM Press

Full text available: pdf(930.61 KB) Additional Information: [full citation](#), [references](#), [index terms](#)



Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright ?2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

**Dialog DataStar**

options

logout

feedback

help

databases

search  
page

titles



## Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

previous  
documentsnext  
documents

order

### USPTO Full Text Retrieval Options

☒ document 851 of 1112 [Order Document](#)

**INSPEC - 1969 to date (INZZ)**

#### Accession number & update

6046140, C9811-6150G-015; 981006.

#### Title

A lightweight architecture for program **execution** monitoring.

#### Author(s)

Jeffery-C; Wenyi-Zhou; Templer-K; Brazell-M.

#### Author affiliation

Dept of Comput Sci, Texas Univ, San Antonio, TX, USA.

#### Source

ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, Montreal, Que., Canada, 16 June 1998.

Sponsors: ACM.

In: SIGPLAN-Notices (USA), vol.33, no.7, p.67-74, July 1998.

#### CODEN

SINODQ.

#### ISSN

ISSN: 0362-1340, CCCC: 1 58113 055 4/98/0006.. (\$5.00).

#### Availability

SICI: 0362-1340(199807)33:7L:67:LAP; 1-G.

#### Publication year

1998.

#### Language

EN.

#### Publication type

CPP Conference Paper, J Journal Paper.

#### Treatment codes

P Practical.

#### Abstract

The Alamo monitor architecture reduces the difficulty of writing dynamic analysis tools such as special purpose profilers, bug detectors, and visualizations. The Alamo monitor architecture extends and generalizes the work done for monitoring in the **Icon** language interpreter by adapting the **execution** model and developing implementation techniques suitable for monitoring compiled programs. Alamo stands for A Lightweight Architecture for Monitoring. The Alamo architecture consists of: (1) an automatic instrumentation mechanism; (2) an **execution** model; (3) abstractions for event selection, multiplexing and composition; and (4) an access library that allows monitors to directly manipulate target program state. These four components are applicable to many compiled and interpreted languages. (19 refs).

#### Descriptors

program-interpreters; system-monitoring; visual-programming.

**Keywords**

lightweight architecture; program **execution** monitoring; Alamo monitor architecture; dynamic analysis tools; **Icon** language interpreter; **execution** model; implementation techniques; compiled programs; automatic instrumentation mechanism; event selection; access library; target program state; interpreted languages.

**Classification codes**

C6150G (Diagnostic, testing, debugging and evaluating systems).  
C6130B (Graphics techniques).  
C6110V (Visual programming).  
C6150C (Compilers, interpreters and other processors).  
C6180 (User interfaces).

**Copyright statement**

Copyright 1998, IEE.

COPYRIGHT BY Inst. of Electrical Engineers, Stevenage, UK

locally as:   search strategy:

Top - News & FAQs - Dialog

© 2005 Dialog

**Dialog DataStar**

options

logout

feedback

help



databases

search  
page

titles

## Document

Select the documents you wish to [save](#) or [order](#) by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

previous  
documentsnext  
documents

order

☒ **document 913 of 1112** [Order Document](#)
**INSPEC - 1969 to date (INZZ)**
**Accession number & update**

5752706, A9724-2843-012, B9712-8220B-114; 971111.

**Title**

A fuzzy logic decision making for multi-module signal validation algorithms.

**Author(s)**
[Erbay-A-S](#); [Upadhyaya-B-R](#).

**Author affiliation**

Dept of Nucl Eng, Tennessee Univ, Knoxville, TN, USA.

**Source**

Proceedings of 1996 ANS International Topical Meeting on Nuclear Plant Instrumentation Control and Human Machine Interface Technologies, vol.2, University Park, PA, USA, 6-9 May 1996.

Sponsors: ANS, EPRI, US DOE, US Nucl. Regulatory Comm. (NRC), Three Human Factors &amp; Ergonomics Soc., IEEE, et al.

In: p.1337-41 vol.2, 1996.

**ISSN**

ISBN: 0-89448-610-1.

**Publication year**

1996.

**Language**

EN.

**Publication type**

CPP Conference Paper.

**Treatment codes**

T Theoretical or Mathematical.

**Abstract**

Reliability of the information collected from a process plant is important for its safe operation and efficient control. Validation of this information may be accomplished by different algorithms, each of them providing a different index of validity. A new approach of interpreting signal validation results from a multi-modular signal validation system is presented in this paper. This uses fuzzy-logic fault-tree methodology. The methodology is based on mapping individual signal validation module results into fuzzy sets in which the domain is the "truthness of the sensor that has failed". Thereafter, a fault-tree methodology is employed to make an overall final decision about the failure of the sensor. The decision making is implemented on a PC-based platform using Microsoft Windows. Fuzzy logic based decision-making results are displayed in "icon" representation, which enables easy recognition of sensor validity at relatively high sampling rates. This approach has been evaluated by application to data from operational pressurized water reactors. (4 refs).

**Descriptors**
[fault-trees](#); [fission-reactor-monitoring](#); [fuzzy-logic](#); [fuzzy-set-theory](#); [pressure-measurement](#); [signal-processing](#).

**Keywords**

fuzzy logic decision making; multi module signal validation algorithms; information reliability; safe operation; fuzzy logic fault tree methodology; fuzzy sets; sensor failure; decision making; Microsoft Windows; PC based platform;

operational pressurized water reactors; pressure measurement.

**Classification codes**

A2843D (Core control and guidance in fission reactors).  
A2850G (Light water reactors).  
A0630N (Pressure measurement).  
B8220B (Nuclear reactors).  
B7320V (Pressure and vacuum measurement).  
B6140 (Signal processing and detection).  
B0170N (Reliability).  
B0250 (Combinatorial mathematics).

**Copyright statement**

Copyright 1997, IEE.

COPYRIGHT BY Inst. of Electrical Engineers, Stevenage, UK

[save](#)

locally as: PDF document



search strategy: do not include the search strategy

[previous documents](#)[next documents](#)[order](#)

[Top](#) - [News & FAQs](#) - [Dialog](#)

© 2005 Dialog

**Dialog DataStar**

options

logout

feedback

help

databases

search  
page

titles



## Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

previous  
documentsnext  
documents

order

### USPTO Full Text Retrieval Options

☒ document 1104 of 1112 [Order Document](#)

**INSPEC - 1969 to date (INZZ)**

#### Accession number & update

4883563, C9504-6160S-003; 950228.

#### Title

An iconic query language for topological relationships in GIS.

#### Author(s)

[Lee-Y-C](#); [Chin-F-L](#).

#### Author affiliation

Dept of Geodesy & Geomatics, New Brunswick Univ, Fredericton, NB, Canada.

#### Source

International-Journal-of-Geographical-Information-Systems (UK), vol.9, no.1, p.25-46, Jan.-Feb. 1995.

#### CODEN

IJGSE3.

#### ISSN

ISSN: 0269-3798, CCCC: 0269-3798/95/ (\$10.00).

#### Publication year

1995.

#### Language

EN.

#### Publication type

J Journal Paper.

#### Treatment codes

P Practical.

#### Abstract

The study of query languages for spatial databases is an active research area. This paper describes a new spatial query language that uses a visual grammar to express topological relationships. It is supplemented by text and icons to handle other spatial and non-spatial queries. A graphical user interface is also developed to provide an interactive environment for composing the iconic query command. To test the language, the interface is implemented on a SUN 4 Workstation and linked to Ingres, a relational DBMS. Preliminary tests show that the iconic query language is more convenient for expressing spatial concepts than conventional textual languages. This is due mainly to the two-dimensionality of iconic languages in contrast with the linear nature of conventional languages. (31 refs).

#### Descriptors

[geographic-information-systems](#); [grammars](#); [graphical-user-interfaces](#);  
[query-languages](#); [relational-databases](#); [visual-databases](#); [visual-languages](#).

#### Keywords

GIS; geographic information system; **icon**; iconic query language; topological relationships; spatial database; spatial query language; visual grammar; graphical user interface; GUI; iconic query command; Ingres; relational DBMS; relational database management system; two dimensional.

#### Classification codes



C6160S (Spatial and pictorial databases).  
C6160D (Relational databases).  
C6180G (Graphical user interfaces).  
C7840 (Geography and cartography computing).

**Copyright statement**

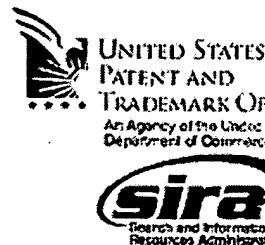
Copyright 1995, IEE.

COPYRIGHT BY Inst. of Electrical Engineers, Stevenage, UK

locally as:   search strategy:

[Top - News & FAQs - Dialog](#)

© 2005 Dialog

**Dialog DataStar**[options](#)[logoff](#)[feedback](#)[help](#)[databases](#)[search  
page](#)

## Titles

To view one or many selected titles scroll down the list and click the corresponding boxes. Then click display at the bottom of the page. To view one particular document click the link above the title to display immediately.

[previous  
titles](#)

Documents 1101 to 1112 of 1112 from your search "**(EXECUT\$ OR COPYRIGHT) AND ICON**" for information added since 19950101:

Number of titles selected from other pages: 14

[deselect all](#)

selected titles

- ☐ **Select All**
- ☐ 1101 [display full document](#)  
1994. (INZZ) My Psion Series 3a pocket computer.
- ☐ 1102 [display full document](#)  
1994. (INZZ) An approach to object-oriented modeling and performance evaluation.
- ☐ 1103 [display full document](#)  
1994. (INZZ) Optimal management of a large computer network with CHIP.
- ☒ 1104 [display full document](#)  
1995. (INZZ) An iconic query language for topological relationships in GIS.
- ☐ 1105 [display full document](#)  
1994. (INZZ) Pascal external functions for Strawberry Tree's "Analog Connection Workbench" on the Macintosh.
- ☐ 1106 [display full document](#)  
1994. (INZZ) Graphical object-oriented programming with LabVIEW.
- ☐ 1107 [display full document](#)  
1994. (INZZ) The effective use of animation in simulation model validation.
- ☐ 1108 [display full document](#)  
1994. (INZZ) Implementing metaphor graphics in an M environment.
- ☐ 1109 [display full document](#)  
1994. (INZZ) Integrated windows-based control system for an electron microscope.
- ☐ 1110 [display full document](#)  
1994. (INZZ) Multimedia learner control and visual sensory-level supports for learning aircraft names and shapes.
- ☐ 1111 [display full document](#)  
1995. (INZZ) Immersion diversions (CD-ROM interfaces).
- ☐ 1112 [display full document](#)  
1994. (INZZ) Wux: Unix tools under Windows.

Selection	Display Format	Output Format	ERA <sup>SM</sup> Electronic Redistribution & Archiving	Action

<input checked="" type="radio"/> from this page <input type="radio"/> from all pages	<input checked="" type="radio"/> Full	<input checked="" type="radio"/> HTML	Copies you will redistribute: <input type="text"/>	<input type="button" value="display"/>
	<input type="radio"/> Free	<input type="radio"/> Tagged (for tables)	Employees who will access archived record (s): <input type="text"/>	<input type="button" value="save"/>
	<input type="radio"/> Short	<input type="radio"/> PDF	<a href="#">Help with ERA</a>	<input type="button" value="print preview"/>
	<input type="radio"/> Medium	<input type="radio"/> RTF		
	<input type="radio"/> Custom			
	<a href="#">Help with Formats</a>			
				<input type="button" value="order"/>
Sort your entire search result by <input type="text" value="Publication year"/> <input type="text" value="Ascending"/>				<input type="button" value="sort"/>

[previous titles](#)

Top - News &amp; FAQs - Dialog

© 2005 Dialog



Welcome United States Patent and Trademark Office

☐ View Selected Items
[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)[SUPPORT](#)

Results for " ((execut\$ or copyright and icon)&lt;in&gt;metadata) "

Your search matched 2 of 1263585 documents. You selected 2 items.

e-mail
 printer friendly

» Download Citations

Display Format:



Citation



Citation &amp; Abstract

Article Information

View: 1-2 | [View Search Results](#)
 >>
» [Learn more](#)

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

## 1. Shrink-wrap license restrictions-preempted?

Stern, R.H.

Micro, IEEE

Volume: 17 Issue: 1 Jan/Feb 1997

Page(s): 75-78

Digital Object Identifier 10.1109/40.566210

**Summary:** Software publishers are fond of shrink-wrap licenses. These notices accompany copies of computer programs and state that by using the program you agree not to do various things that the publisher does not want you to do. Typically, for example, the p.....

[AbstractPlus](#) | [References](#) | Full Text: [PDF](#) IEEE JNL

## 2. Copyright

Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on

Volume: 1 16-19 Nov. 2004

Page(s): ii- ii

**Summary:** Not available.....Full Text: [PDF](#) IEEE CNFView: 1-2 | [View Search Results](#) | [Back to top](#)
 Indexed by  
 Inspec\*

[Help](#) | [Contact Us](#) | [Privacy & Security](#) | [IEEE.org](#)

© Copyright 2005 IEEE – All Rights Reserved